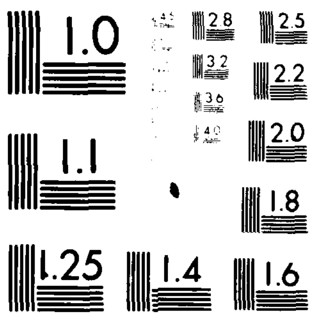
AD-A113 004 ENVIRONMENTAL RESEARCH AND TECHNOLOGY INC LEXINGTON MASS F/G 8/2
MINI RASTER-TO-VECTOR CONVERSION.(U)
SEP 81 R K CRANE DAAK70-81-C-0025
UNCLASSIFIED ERT-P8002 ETL-0269 NL

1 OF 1
AD
A113004

END
DATE
FILMED
5-82
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

ETL-0269

AD A11 3004

Mini raster-to-vector conversion

Robert K. Crane

Enviromental Research and Technology, Inc.
696 Virginia Road
Concord, MA 01742

SEPTEMBER 1981

U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060

E T L

ETL-0269

(12)

MINI RASTER-TO-VECTOR CONVERSION

Final Report

ERT Document No. PB002
Contract No. DAAK70-81-C-0025

September 1981

Prepared for

Department of the Army
U.S. Army Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060

Prepared by

Robert K. Crane

Environmental Research & Technology, Inc.
696 Virginia Road
Concord, Massachusetts 01742

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0269	2. GOVT ACCESSION NO. AD-A112 004	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Mini Raster-to-Vector Conversion		5. TYPE OF REPORT & PERIOD COVERED Contract Report
		6. PERFORMING ORG. REPORT NUMBER B002 (Final)
7. AUTHOR(s) Robert K. Crane		8. CONTRACT OR GRANT NUMBER(s) DAAK79-81-C-0025
9. PERFORMING ORGANIZATION NAME AND ADDRESS Environmental Research & Technology, Inc. 696 Virginia Road Concord, Massachusetts 01742		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 3203 MT34
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060		12. REPORT DATE September 1981
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

raster
vector
contour
topographic map

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A raster-to-vector conversion program was developed for the Defense Mapping Agency (DMA). The program operates on run length coded data from a SCI-TEX scanner. It operates in a 16-bit Digital Equipment Corporation PDP 11/60 minicomputer with limited core storage, less than 64 k-bytes. The program was tested and timed using data supplied by DMA. It vectorized 4879.8 linear inches of contour data in 1.19 hours at 0.005 inch resolution and 1.14 hours at 0.010 inch resolution. The program also developed a number of attributes and contour intersection directories useful for automated editing of the vector data.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This document is generated under Contract DAAK70-81-C-0025 for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia 22060 by Environmental Research & Technology, Inc., Concord, Massachusetts 01742. The Contract Officer's Representative was Mr. Douglas R. Caldwell.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Program Objectives

Environmental Research & Technology, Inc. (ERT) developed a raster-to-vector conversion algorithm as an element in an automated cell detection scheme for processing weather radar data (Crane, 1979). The objective of this contract with the U.S. Army Engineer Topographic Laboratories (ETL) was to implement the previously developed algorithms on a Digital Equipment Corporation (DEC) PDP 11/60 computer at the Defense Mapping Agency (DMA) and demonstrate its use for vectorizing topographic maps. Specifically, the work was divided into four tasks:

1) modify an existing program to operate on a 16-bit mini-computer of limited (128 k-byte) memory capacity;
2) develop initial editing algorithms to remove spurs from contour lines;
3) test algorithms; and
4) implement and conduct a timing test of the finished program on the PDP 11/60 at DMA using data provided by DMA obtained from the SCI-TEX scanner.

The final product of this contract is the results of the timing tests (Task 4).

1.2 Summary of Results

The raster-to-vector conversion algorithm developed by ERT is efficient in its use of computer core storage because only two raster scan lines of data are stored at a time. It is also rapid because a minimum number of scan line-to-scan line comparisons are employed in the contouring operation. Its major advantage over other contouring algorithms is its use of attributes to further characterize each contoured region (or line). The primary objective of this contract with ETL is the demonstration and timing of the basic contouring algorithms. In developing the computer programs for installation on the PDP 11/60 at the Defense Mapping Agency, a minimum number of attributes were maintained for future use in automated editing

of the vector output. A discussion of the utility of attributes for automated editing is presented in the fourth section of this report.

Two PDP 11/60 computer programs were developed for ETL under this contract. The first, GETTAP, reformats the SCI-TEX output tape data for storage and ready access on the PDP 11/60 disk system; the second, DMA10, performs the raster-to-vector conversion and spur identification operations. The timing of the second program on the PDP 11/60 for an entire 15' x 15' topographic map provides the final results of this contract. The results of the timing tests for the DMA supplied map (SHIRAZ test sheet, second version) are listed in Table 1. Results are provided for vector output at two raster line spacings, 0.005 inches (5 mil) and 0.01 inch (10 mil). The total vector line length (sum of the lengths of all the vectors) was 4879.8 linear inches. A total of 33103 identifiable contour line segments were detected separating 15220 nodes.

The modification of the basic contouring algorithm for use on a 16-bit minicomputer of limited core size was accomplished at ERT using one of their Data General Eclipse 250 computers. Initial algorithm testing was accomplished using SCI-TEX output from a partially edited topographic map supplied by DMA (SHIRAX test sheet, first version). Due to limited funding, the entire map was not processed on the ERT computer. Processing was accomplished for six subregions of the map which were selected to illustrate and test different editing problems. A comparative set of results for the entire map using a Data General Eclipse and the DMAHTC PDP 11/60 are listed in Table 1. The Eclipse values were estimated from the results for the six subregions as discussed in Section 3 of this report.

1.3 Computer Programs

Two computer programs were generated for the PDP 11/60 minicomputer under this contract, GETTAP and DMA10. An additional program DMA9 was also developed for algorithm testing on the ERT Eclipse 250 mincomputer. The raster-to-vector programs differ to allow the PDP 11/60 version to take advantage of the 4-byte integer word capabilities of Fortran IV Plus in processing the large numbers of contour segments and modes expected for the entire map. The ERT version, DMA9, also differed from the final

TABLE 1

DMA10 TIMING RESULTS FOR DMA SUPPLIED SCI-TEX DATA

Output Spacing (mils)	Total Length of Vectors (linear inches)	Wall Clock Time (hours)	CPU Time (hours)	Number of Output Blocks[†] (numbers)	Mini-Computer System
5**	4879.8	1.19	N/A	19877	DMA PDP 11/60
10	4879.8	1.14	N/A	10624	DMA PDP 11/60
10	4880*	4.2*	0.58*	N/A	Eclipse 250

*estimated for the full SHIRAZ test map from DMA9 runs on the ERT Eclipse 250

**NTEN = 5 or 10 in DMA10

[†]512 byte blocks

version of the program, DMA10 by generating and outputting four additional attributes for each contour line segment. These attributes were used for intermediate display generation and were not required for the final version of the program.

Initial information provided by DMA specified the computer core storage at 128 k-bytes (Task 1). At the time of installation, it was found that only 64 k-bytes were available for a single user and, in practice, less was available because of the system routines and buffers in each user area. The available core limitations forced the restructuring of some of the programs and a reduction in the number of attributes stored for each contour segment for future editing.

Initially, the internal record keeping was designed to utilize SORT routines for reorganizing the vector data for rapid display as concatenated vector arrays. The discovery that DMA did not acquire SORT routines for the PDP 11/60 system forced a revision of the internal record keeping. The revised data management system utilized directories of the line segments that intersect at each node so that, with the direct access disk storage system, each line segment can be identified and recovered for separate display. A third program, FETCH5, was written by ERT using company funds to reorganize the vector data to provide concatenated vector arrays in the Automated Graphics Display System (AGDS) format used by the Defense Mapping Agency. ERT is making that program available to DMA for reformatting the output for display on their system. The program will not be documented in this report because it was not developed under this contract. It is being made available to DMA to provide a means to graphically observe the output from the raster-to-vector conversion program.

1.4 Organization of Report

This report presents the results of the timing runs of DMA10, the program developed by ERT to rapidly transcribe raster data into a vector line segment format. The timing results are given in Table 1. The raster-to-vector conversion program is described in Section 2, a detailed demonstration of its operation is provided in Section 3, and recommendations for the automation of the editing process currently required to complete the raster-to-concatenated vector conversion are given in Section 4. Program listings for GETTAP and DMA10 and operating instructions for the programs on the DMA PDP 11/60 are given by Gustafson (1981).

2. DMA10, THE MINI-RASTER/VECTOR CONVERSION PROGRAM

The mini-raster-to-vector conversion program reads the reblocked SCI-TEX data from the disk (Directory in F1, run length coded data in F2), performs the scan line-by-scan line association and outputs the node and contour line directories, contour line vectors, and contour line attributes. Figure 1 presents the flow chart for this program. The program performs two key tasks, (1) raster scan line-by-raster scan line association to follow a contour from one raster scan line to the next, and (2) directory preparation to keep track of each of the contour lines (identities) that intersect at a node. The program maintains a list of attributes for each line segment (Table 2) and directories to locate the line segment identifier, map locations, and attributes for each node (Table 3). The output data include the directories, the attributes, and the end points (map locations) for each vector (Table 4).

The heart of the contouring operation is the association process (Figure 2). Contouring is performed using only two raster scan lines at a time. The input from the SCI-TEX scanner are run length coded positions (x) along a raster scan line (at y). The coded positions are for the edges of each contour line. The program generates, from the run length coded data, arrays of left (start) edge locations (x) and right (end) edge locations (x) for each contour line (event). These data are stored in the IC1 (start) and IC2 (stop) arrays using an offset addressing scheme so that the data from both the current and prior lines are stored in the upper and lower (or vice versa) halves of the same array. Data for each current raster line are stored over the data from the previous prior raster line and the addressing changed to minimize the shuffling of data in core as each current line is changed to prior line status when a new current line is read by the program.

Association proceeds by comparing the start and stop boundaries of each event (Figure 2). If an overlap occurs, within event boundaries of one or more contour line segments, between current and prior raster scan lines, the lines are associated. If the overlap is between a single event on the current and a single event on a prior line, the contour line is continued (termination code value of 1, see Table 5). If multiple contour line segments from the prior raster scan line are associated with one (or more)

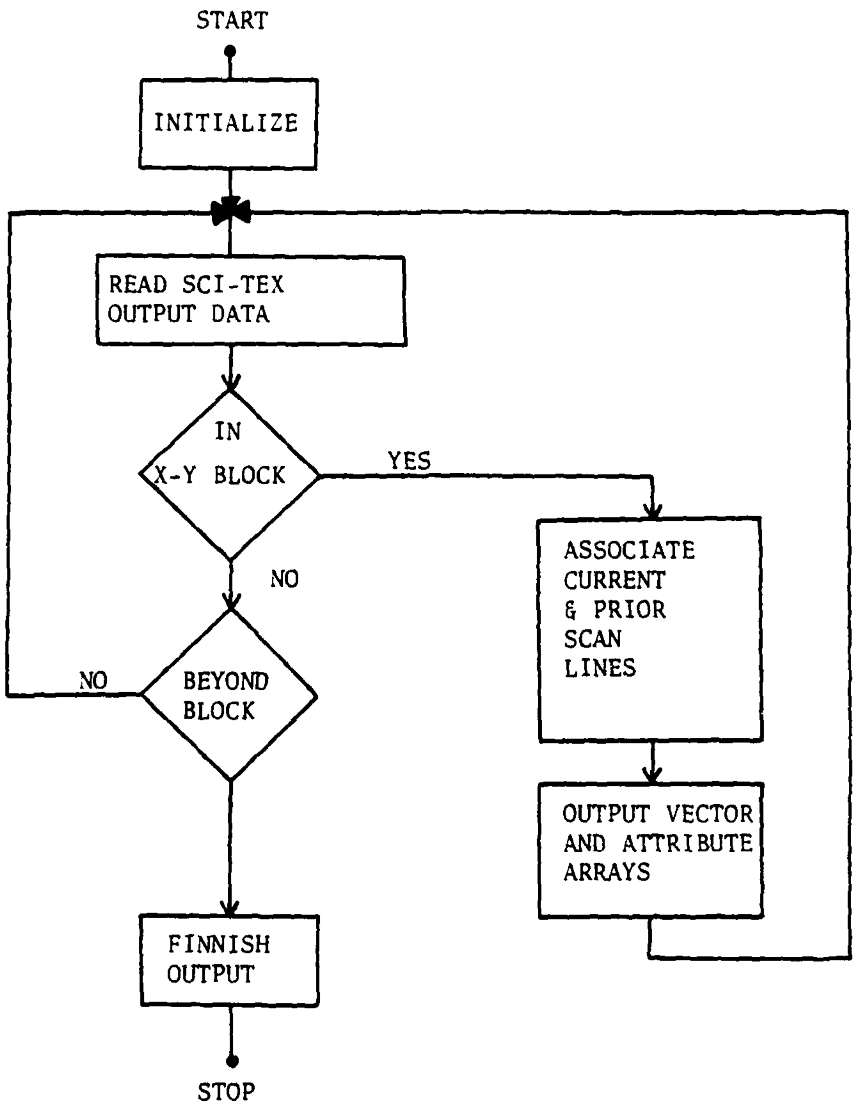


Figure 1 DMA10 Flow Chart

TABLE 2

CONTOUR LINE SEGMENT ATTRIBUTES

Name	Internal Variable	Location in Output Array (4-byte words)	Function
Line Segment Identifier	NEXTID KNID NNID IDSLOT*	-	Counter/line segment identifier
Slot	IID KID IC3	-	Location of attributes for line segment in internal arrays
X	IATR3*	-	Vector end point, stored until next write
Y	TATR4*	-	Vector end point, stored until next write
line length	IATR7*	-	Running length of contour line segment
first vector location	IAPB*	4	Location of first vector end point in output, PT file
Start Node	IATR1*	1	Identifier of starting node (or map boundary) for line segment
Stop Node	-	2	Identifier of ending node for line segment
Line Type	LTT	3, 1st 2 bytes	Line Segment type, -1 identifies a spur, see Table 6
End Code	ITYPE	3, 2nd 2 bytes	Identifies the way a line segment terminates, see Table 5

(internal attribute arrays

TABLE 3

DIRECTORIES

Name	Disk File	Unit	List	Contents
NODA	NODE	10	NODD by node	start location for list of line segment identifiers for node
NODD	NDIRECT	11	line segment identifiers	list of line segment identifiers for node
KNIDR	KDIRECT	12	attribute address by line segment identifier	start location in IA list of attributes for line segment

TABLE 4

OUTPUT FILES

Name	Disk	File	List	Contents
IA	AT	2	attributes by line finish order	attributes for line segment
IP	PT	1	vector line segments	position (x,y) and line segment identifier for the starting end of each line segment

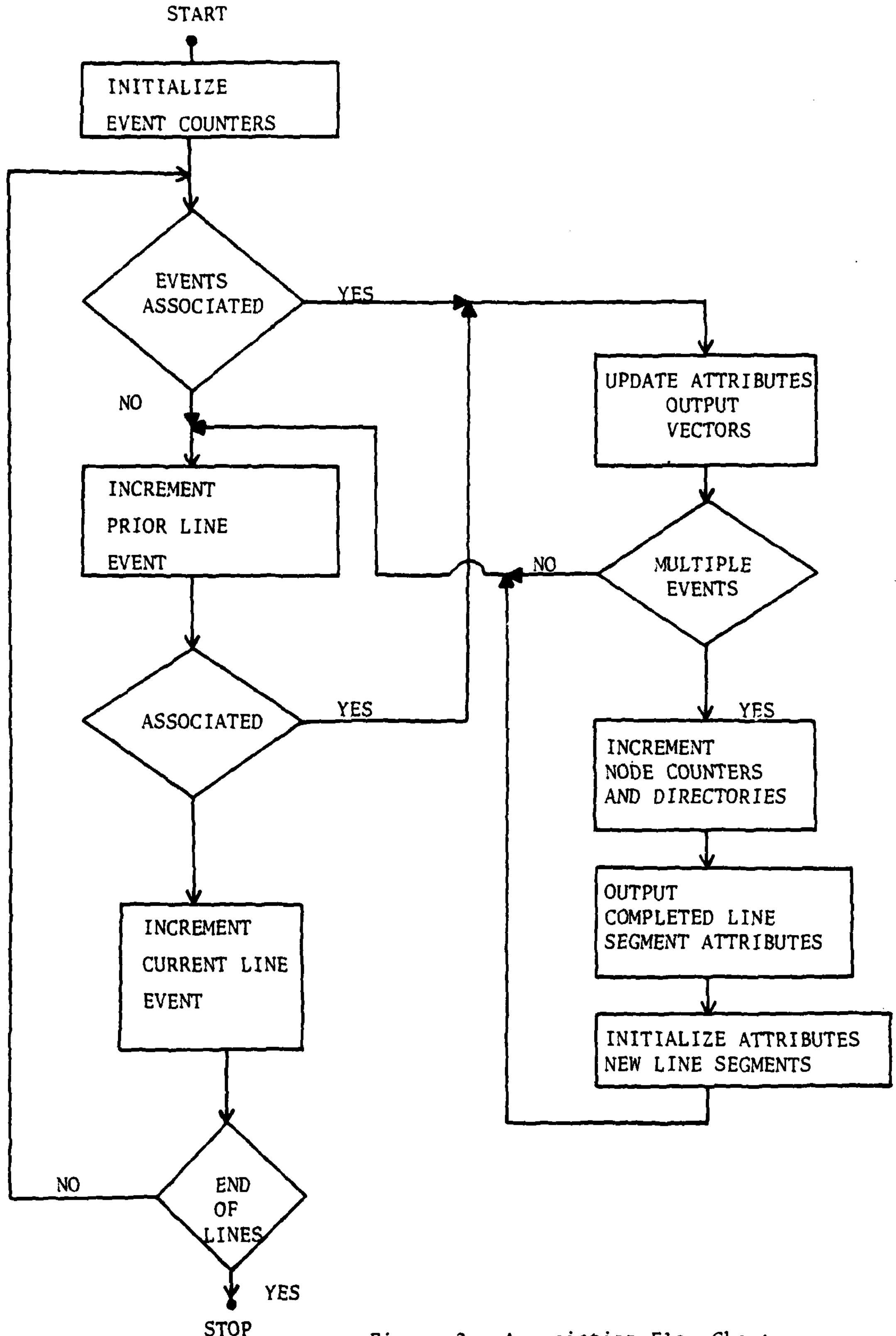


Figure 2 Association Flow Chart

TABLE 5

LINE SEGMENT TERMINATION CODE

Value	Condition
1	continuation, does not terminate
2	terminates left side of "y" (at a node)
3	terminates right side of "y" (at a node)
4	terminates right side of "y" (more than 2 terminations at node)
5	terminates top of "λ" (bell) (at a node)
6	line segment stops (not at a node)
7	line segment stops at lower (y) boundary

line segments on the current line, a node is encountered causing the line segments to be terminated at a "y" and new contour line segments to be initiated below the "y". If one contour line segment from the prior raster scan line branches into more than one line segment on the current raster scan line, it is terminated at a "λ" (bell) on the prior raster scan line. If no association is possible, the contour line segment is terminated if it last occurred on the prior raster scan line, or a new line segment (unassociated) is declared if it first occurs on the current raster scan line.

Valid contour lines both originate and terminate on either the boundaries of the map or at nodes. The nodes are identified by a node counter which is incremented for each new node. Origination or termination on a map boundary is identified by a negative value for the node identification attribute for the contour line segment. Short line segments which do not terminate (or originate) at a boundary or a node are identified as spurs. Open lines (not terminating or originating at a boundary or node) are only classified as spurs if their length is less than 0.02 inches. The line length criterion may be changed when the program is compiled. Although the contour line segment length is maintained as an internal attribute, it is not output; only the line type codes (Table 6), for which line length is one criteria, are output as an attribute.

The information about the nodes at each end of a line segment are output in the attribute list for a line segment. The information about the contour line segments which originate or terminate at a node are maintained in separate, direct access file directories (Table 3). The vector output for each contour line segment is stored in the vector output data file (Table 4). Each vector output record contains the x,y position of an end point, the contour line segment identifier, and the line termination code. Concatenated vector arrays may be formed by selecting from the vector output data file all the points for the desired contour line segment identifier. Initial and intermediate points along the contour line are identified by a continue termination code (value of 1); the end of each vector string by a termination value (>1). To minimize search time for a vector, the data block containing the first vector position record is recorded as an attribute. The directories allow ready access to the attributes when the contour line identification number (or end point node) is specified.

TABLE 6

CONTOUR LINE SEGMENT TYPE

Value	Type
-1	spur
1	short connecting segment (between 2 nodes)
4	open line segment (does not terminate at a node)
5	closed line segment (between 2 nodes)

3. DEMONSTRATION OF THE RASTER-TO-VECTOR PROGRAM

3.1 Operating Time

The Defense Mapping Agency (DMA) supplied ERT with two edited versions of the output from a SCI-TEX scan of a topographic map identified as the SHIRAZ test sheet. The first version was from a partially edited map; the second version, employed for testing at DMA, was from a completely edited map. Due to funding limitations, only small segments of the first version map were processed at ERT for algorithm testing purposes. Six different test areas were selected, ranging in size from 1 to 10 square inches in area, to present as broad a range of processing problems as possible. Selection criteria included contour line density and line length and mapping conventions such as deleted or merging contour lines.

3.1.1 Data General Eclipse 250

Initial algorithm tests and computer program development were performed at ERT using an Eclipse 250 minicomputer. Each of the six test regions were processed by the Eclipse version of the raster-to-vector conversion program, DMA9. Processing time details are given in Table 7 and are represented graphically in Figure 3. One of the outputs from the program is the total contour line length processed in the block or area. The line lengths for each area are listed in Table 7 and used for the estimation of the total operation time to be expected for a map with a specified total line length. Using a 11,000 linear inch map as an example of a high line density map, the estimated running time (CPU) for a Data General Eclipse 250 minicomputer is 1.3 hours. It is important to note that the Eclipse was operated as a multi-task system and the elapsed time depended heavily upon the number of users. Extrapolation to elapsed time for a large map was made for lightly loaded conditions (a multiplier of 7.11). The estimated elapsed time was 9.24 hours for a 11,000 linear inch, high density map and 6.7 hours for a 8,000 linear inch, average density map.

The running time depended both on the number of linear inches of contour lines and the position of the area in the original map. The input data were read sequentially to find the starting raster scan line

TABLE 7

DMA9 RUNNING TIME TESTS

Area	X		Y		DMA 9		Contour Lengths (linear inches)
	Min (in)	Max (in)	Min (in)	Max (in)	Elapsed Time (sec)	CPU Time (sec)	
0	8.099	9.119	2.301	[illegible]	223	31.36	21.3
1	8.241	9.189	9.351	[illegible]	1795	101.97	16.7
2	21.699	22.699	2.291	3.290	2720	55.39	52.3
3	17.099	18.099	10.551	11.550	1134	130.05	40.8
4	8.099	12.099	2.301	4.600	4450	92.18	209.9
5	22.699	23.199	3.221	7.201	1742	159.30	35.9

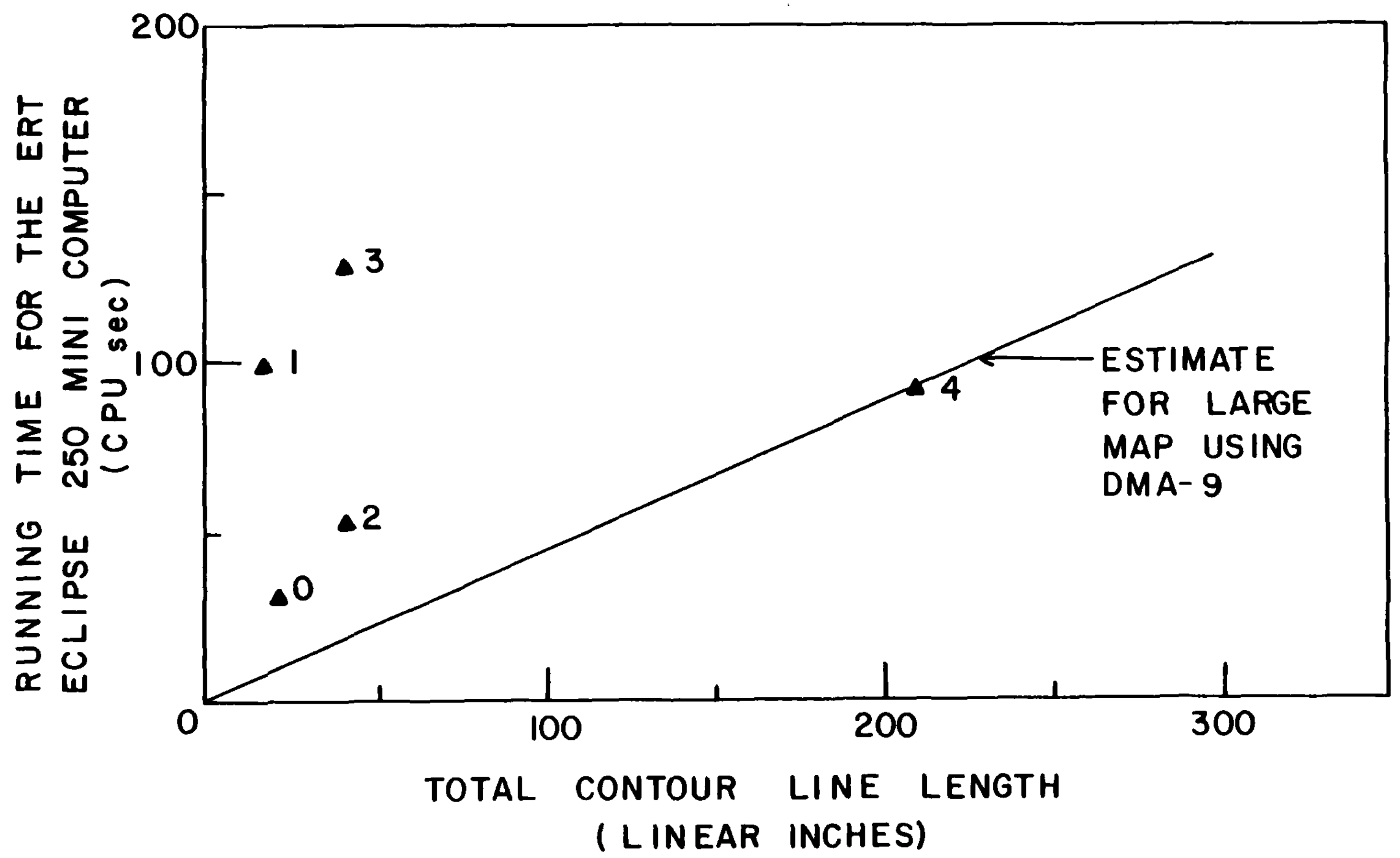


Figure 3 Processing times for various map segments

for a block and along a scan line in a block to locate data within an area. The reading process took a significant amount of both CPU and elapsed time. The results for area 4 were used for extrapolation because they had the smallest ratio of tape reading time to contour processing time. Area 3 was furthest into the map both in raster scan lines (y) and along the line. It and area 1 (large starting y) had the highest ratio of tape reading time to contour processing time.

3.1.2 Digital PDP 11/60

After testing at ERT, the raster-to-vector conversion program was modified to run on the DMANTC PDP 11/60 minicomputer. The PDP 11/60 version, DMA10, was tested using the second version (completely edited) map, on two of the test areas. Timing results are given in Table 8. CPU time was not available on the PDP 11/60 so only elapsed time values are given. The elapsed time was further broken down to be the amount of time required to read through the data to the specified test block and the time actually spent contouring the data. Processing times for the sorting and reformatting routine, FETCH5, are also presented. Two operating modes for FETCH5 were employed during the timing tests; (1) sort and collate the contour segments and nodes to produce complete contour lines; and (2) reformat the complete contour line data into AGDS format. Only the final run listed in Table 8 was performed under the second operating mode. Timing statistics for the reformatting routine, GETTAP, are not available but should be approximately equivalent to the time required to read the data from tape.

3.2 Contour Line Generation

The blocks or areas processed for display are shown superimposed on the original topographic map in Figures 4 (Areas 0 and 4), 5 (Area 1), 6 (Areas 2 and 5), and 7 (Area 3). The output displays from the raster-to-vector conversion program (DMA9) are presented in Figures 8 through 13 for Areas 0 through 5 respectively. The displays were enlarged to illustrate the operation of the raster-to-vector conversion algorithms.

Figure 8 for Area 0 reproduces the detail evident in the original map. The index (heavy or thick) lines visible in the original map (Figure 4) are labeled in the raster-to-vector output map. These lines are not

TABLE 8

DMA10 AND FETCH5 RUNNING TIME TESTS

Area	Resol-ution (mils)	X		Y		ELAPSED TIME (sec)			Total Contour Lengths (linear inches)
		Min (in)	Max (in)	Min (in)	Max (in)	Read	DMA10	FETCH5	
0	10	8.100	9.120	2.300	3.320	198.9	117.9	12.7*	21.0
4	10	8.100	12.100	2.300	4.600	173.3	307.5	161.4*	191.5
Full Map	10	0.000	32.767	0.000	32.767	0	4116.3	3147.6*	4879.8
Full Map	5	0.000	32.767	0.000	32.767	0	4282.5	65810.9	4879.8

*FETCH5 without AGDS reformatting

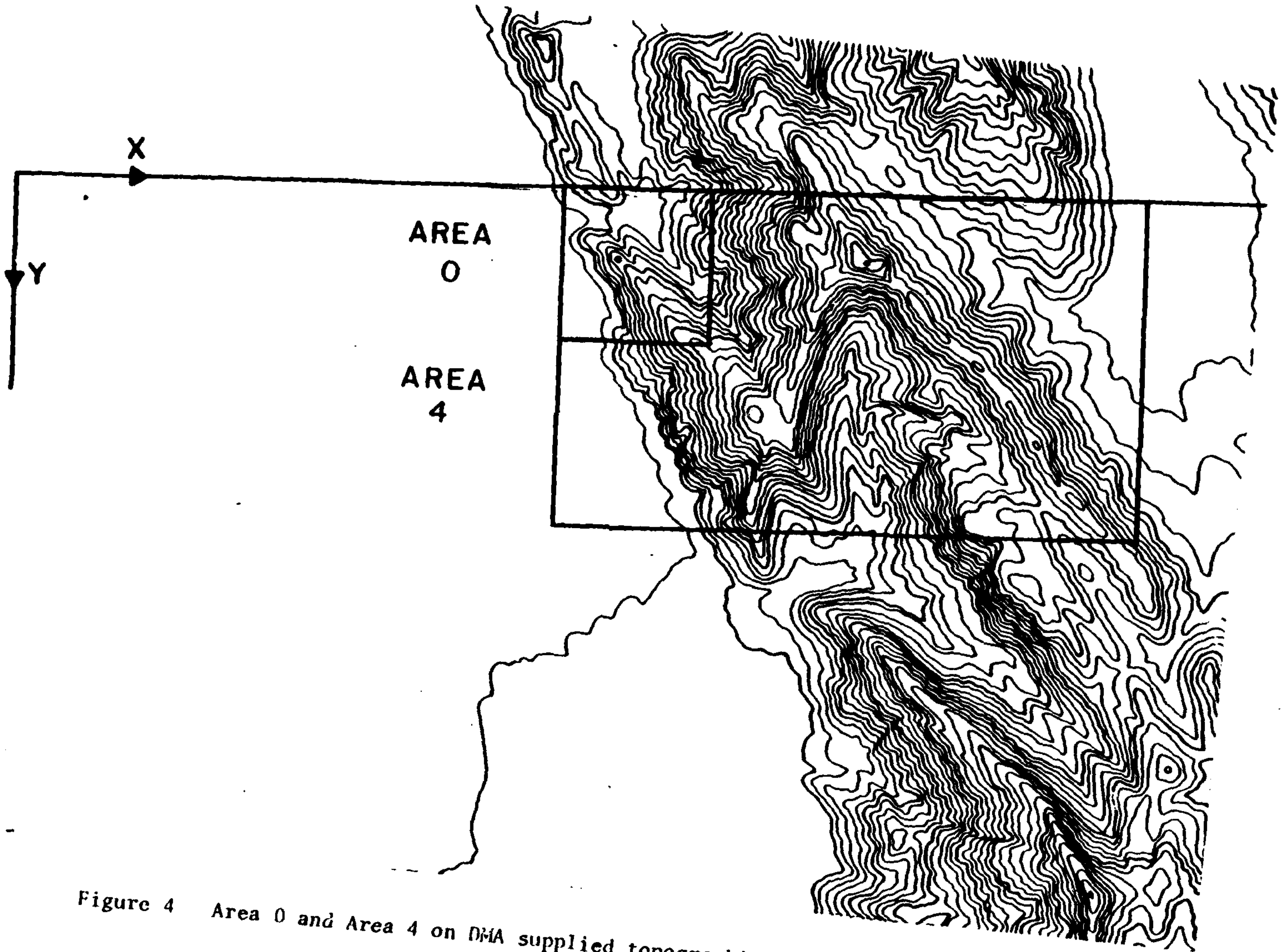


Figure 4 Area 0 and Area 4 on DMA supplied topographic map

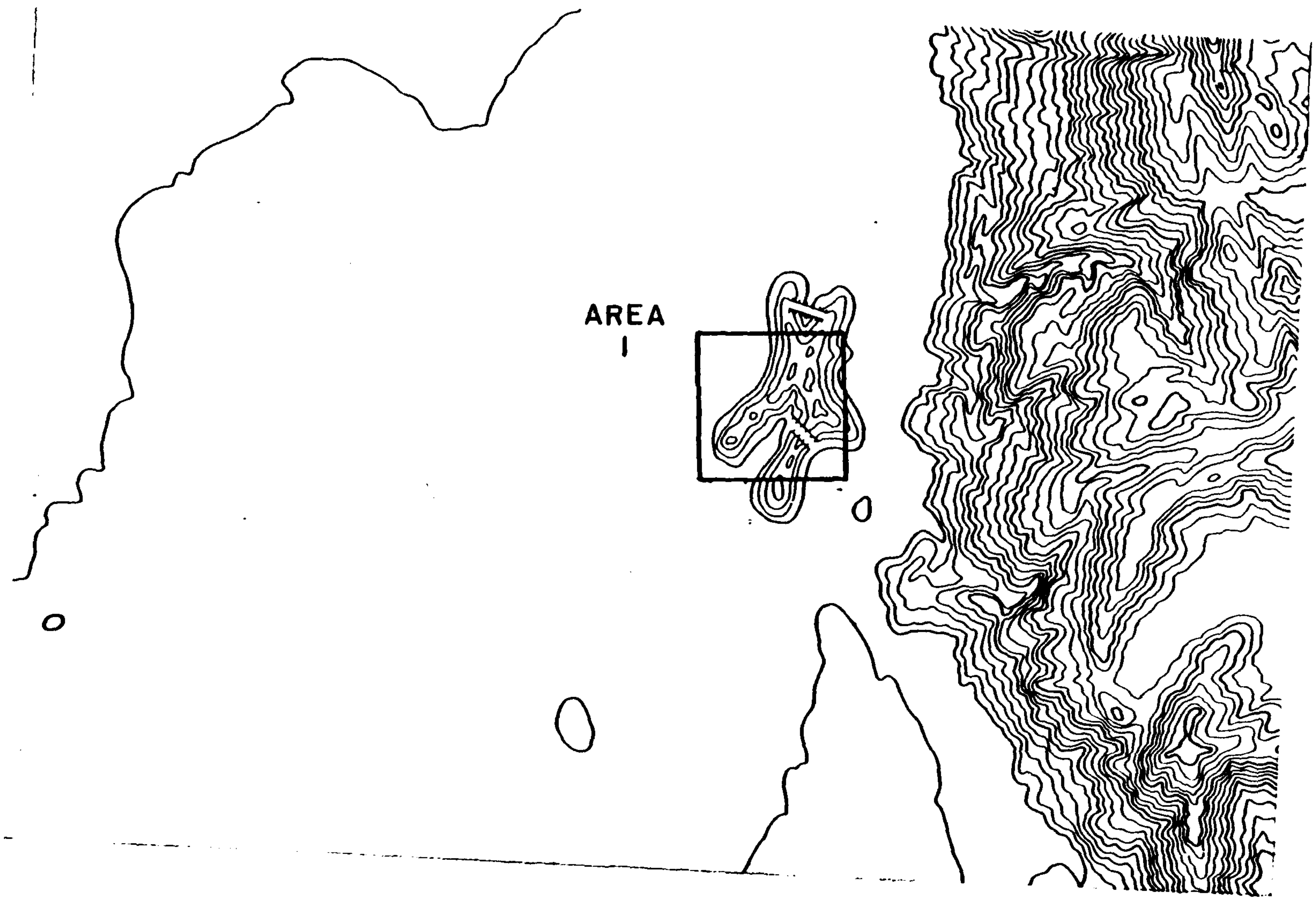


Figure 5   Area 1 on DMA supplied topographic map

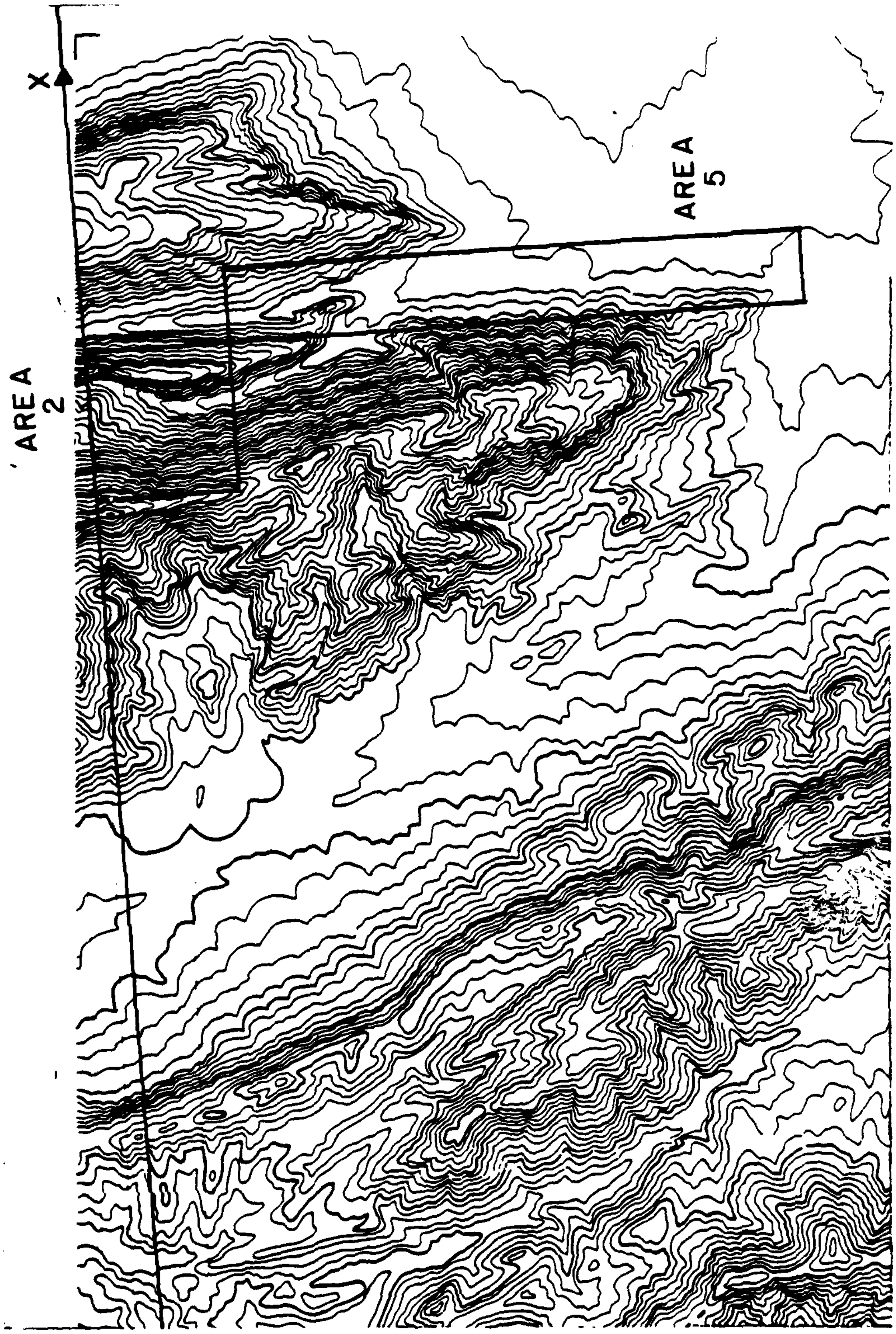


Figure 6 Area 2 and Area 5 on DMA supplied topographic map

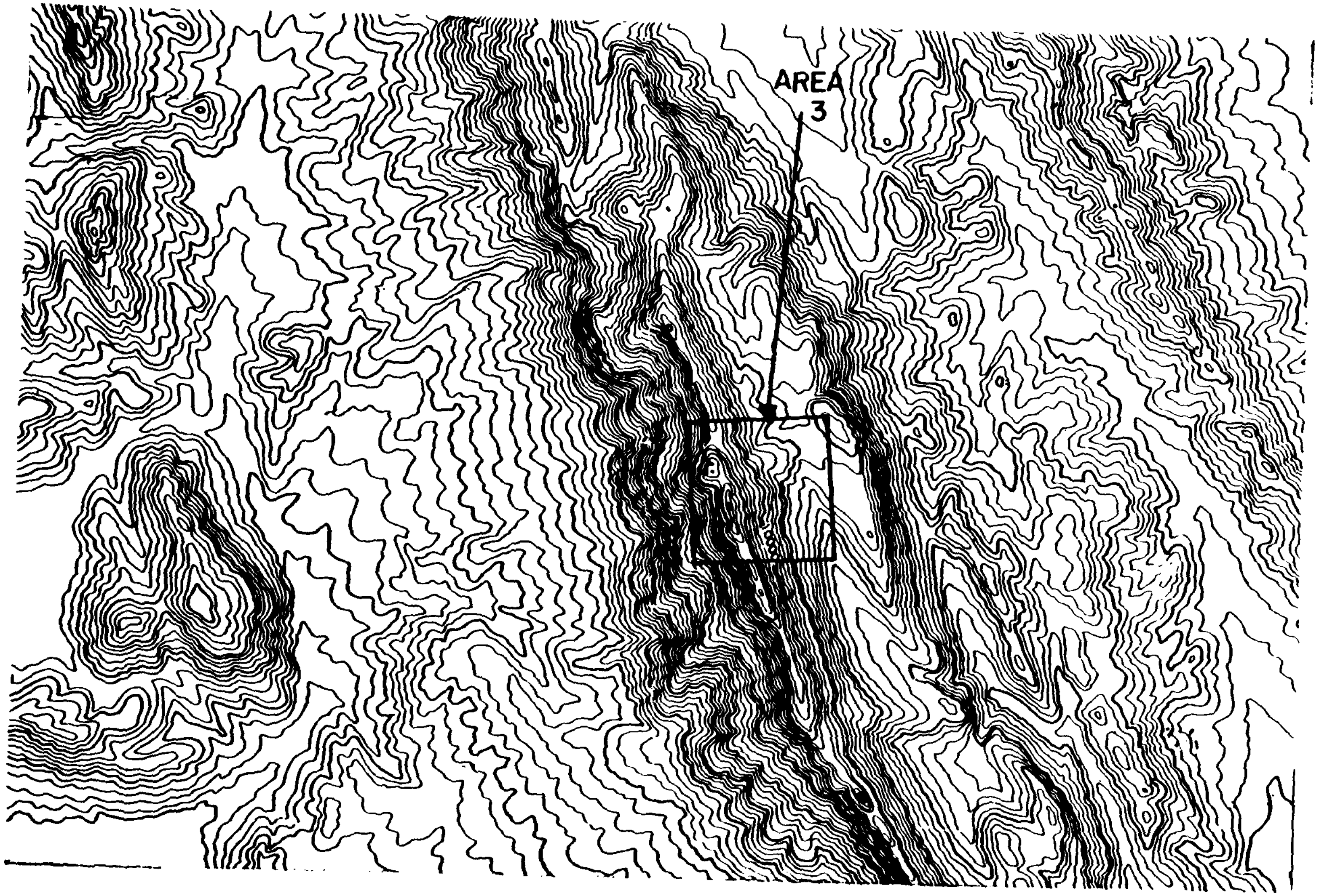


Figure 7 Area 3 on DMA supplied topographic map

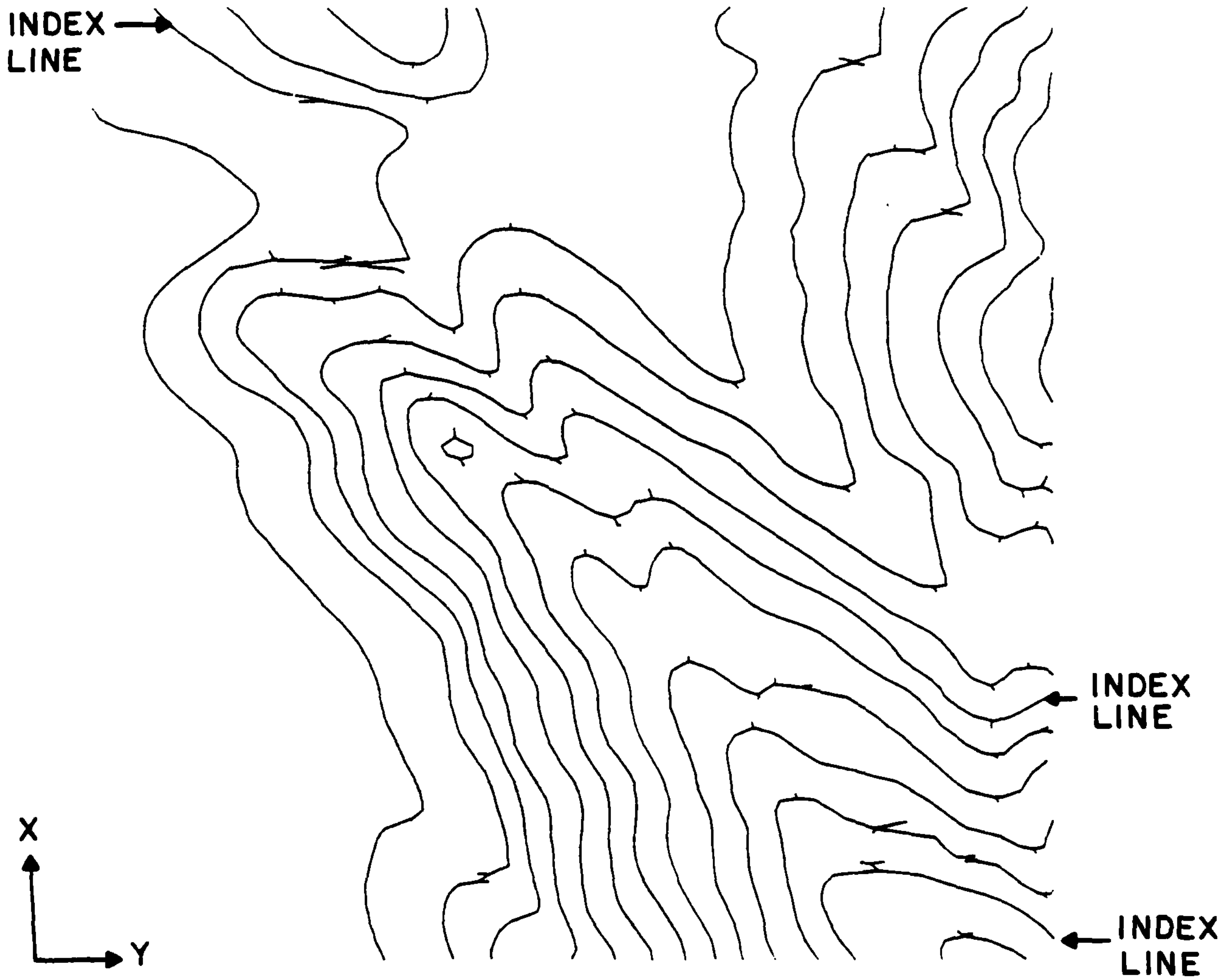


Figure 8   Output from raster-to-vector conversion program for Area 0

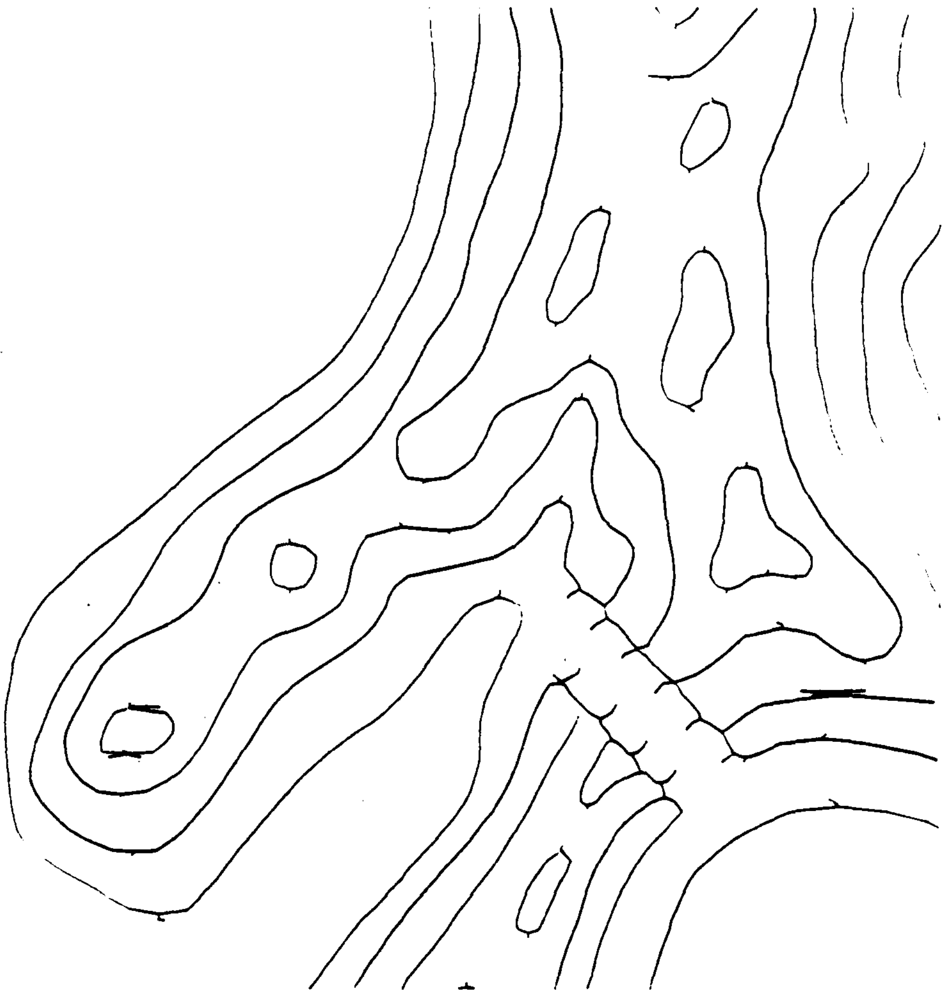


Figure 9 Output from raster-to-vector conversion program for Area 1

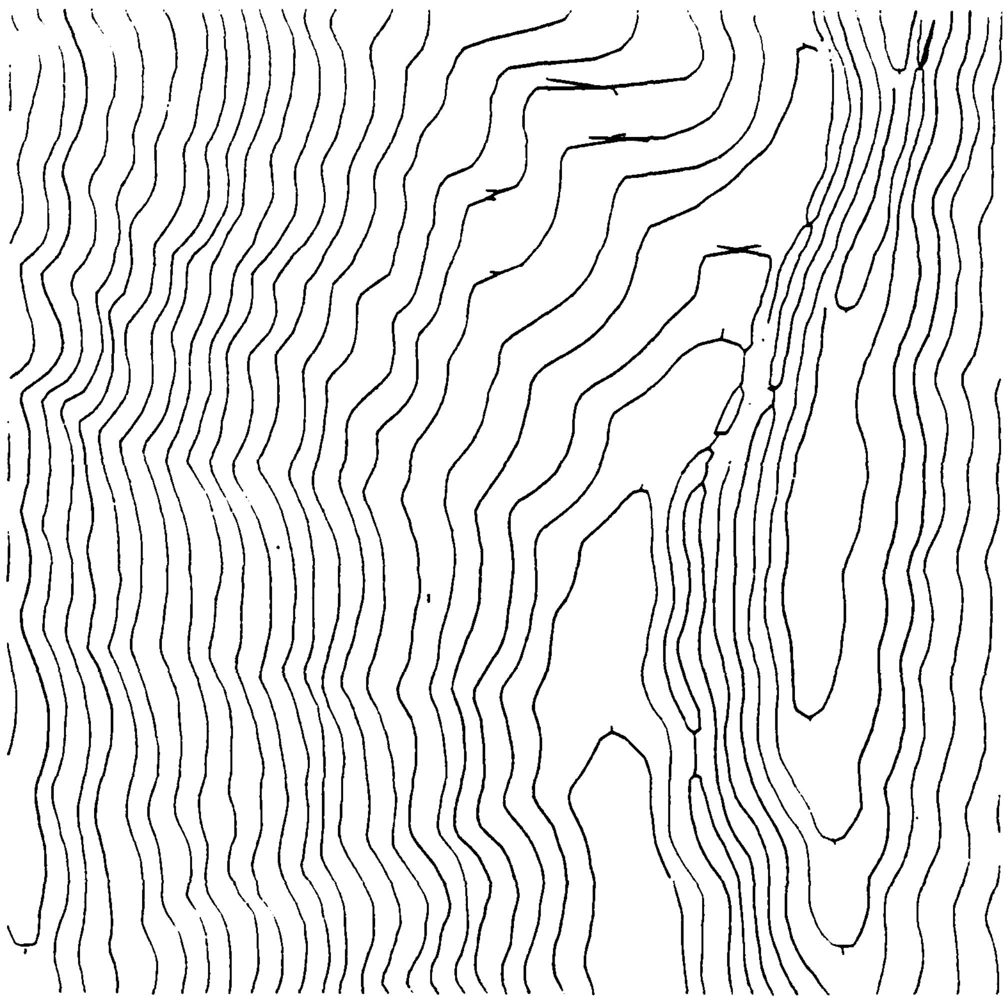


Figure 10 Output from raster-to-vector conversion program for Area 2

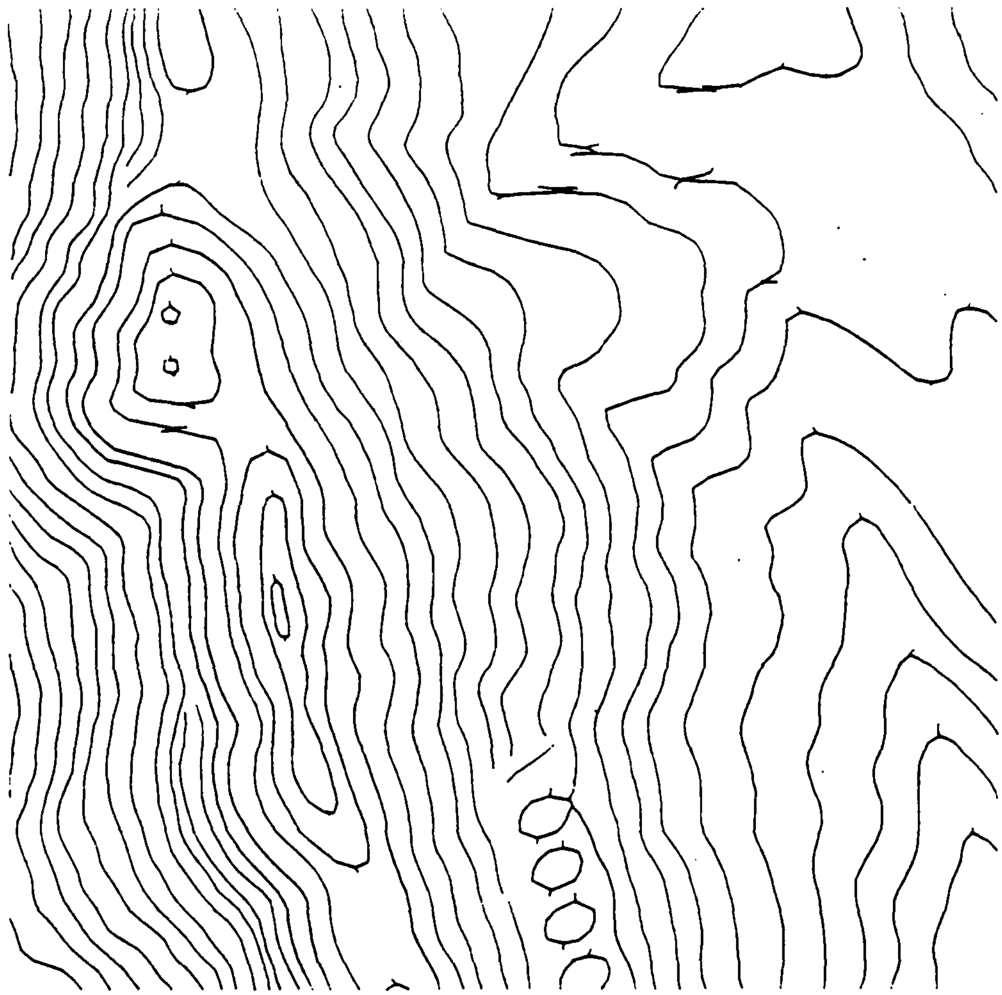


Figure 11 Output from raster-to-vector conversion program for Area 3

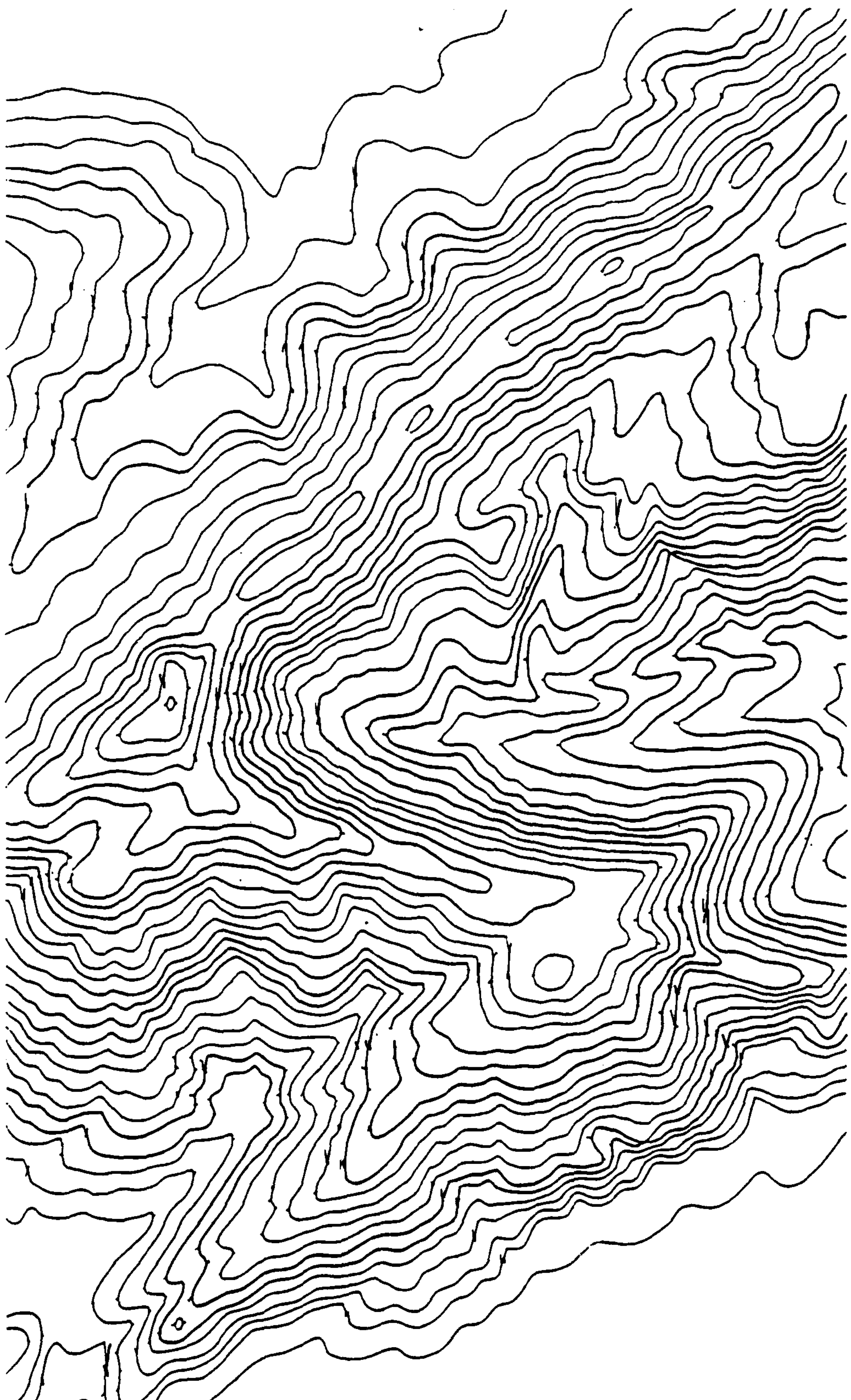


Figure 12 Output from raster-to-vector conversion program for Area 4

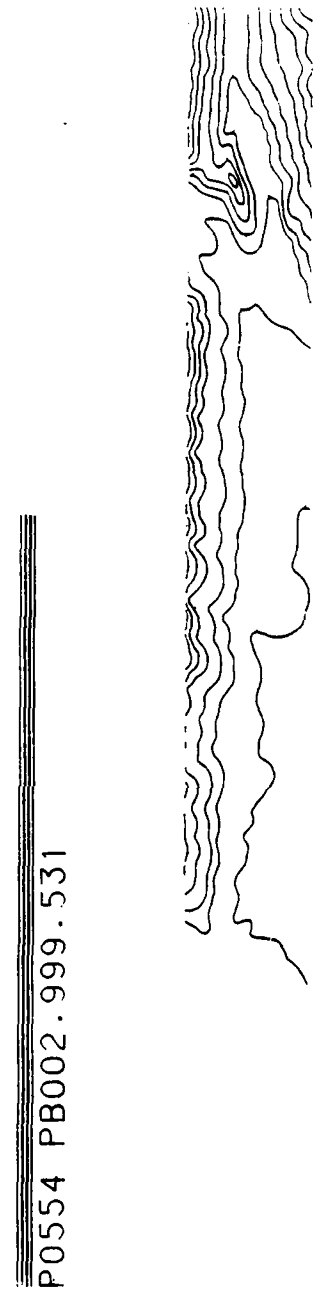


Figure 13 Output from raster-to-vector conversion program for Area 5

automatically recognized in the current version of the program but could be with minor modifications to the program. An earlier version of the program recognized index lines, but this feature was removed because of the limited available core storage in the PDP 11/60 and a desire to analyze the entire map in one pass through the computer program.

Spurs are generated by the line-by-line contouring algorithm because each raster scan line crossing of a contour line is thinned (only the mid point between the right and left edge is saved) to output a single vector for each contour line segment. This process generates spurs at the upper and lower edges of "0"s or at inflection points of the contour lines. Most of the spurs are very short and easily recognized by the computer program. The spurs for Area 0 are displayed in Figure 14. The program identified and labeled all the spurs (type-1 line, see Table 5) but one, a longer open line extending from the left most index line (labeled as a type 4 line, see Table 6). The longer line not identified as a spur is displayed in Figure 15. Experience to date has shown that the longer spurs are associated with the index lines and that all the spurs may be removed by using a longer length criterion for the detection of spurs from index lines.

Two types of contours found in Area 0 required no further editing. They are the isolated closed contours illustrated in Figure 16 and the contours which terminate at the map boundaries displayed in Figure 17. At the current state of automated editing, a number of contours need additional, manual analysis. These contours are displayed in Figure 18. Two types of contours are displayed in this figure, a complex (multispur) contour line which intersects only one boundary and contour lines which fork with both lines (one a spur) touching the boundary .

Area 4 included Area 0 in one corner and provided an example of more complex situations requiring further editing. Figure 19 displays the spurs detected for Area 4, Figure 20 displays the closed contours, and Figure 21 displays the contours which originate and terminate at boundaries of the map area. It is noted that the contours displayed in Figure 18 which needed further editing are displayed in Figure 21 as requiring no further editing. The spurs were successfully identified when the larger area was analyzed. Figure 22 presents the open contour lines (type 4, Table 6), some of which are spurs on index lines (labeled) but several

Figure 14 Spurs in Area 0

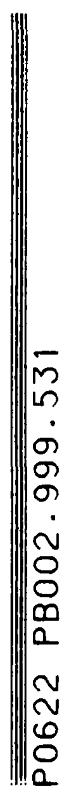


Figure 15 Open line segment in Area 0

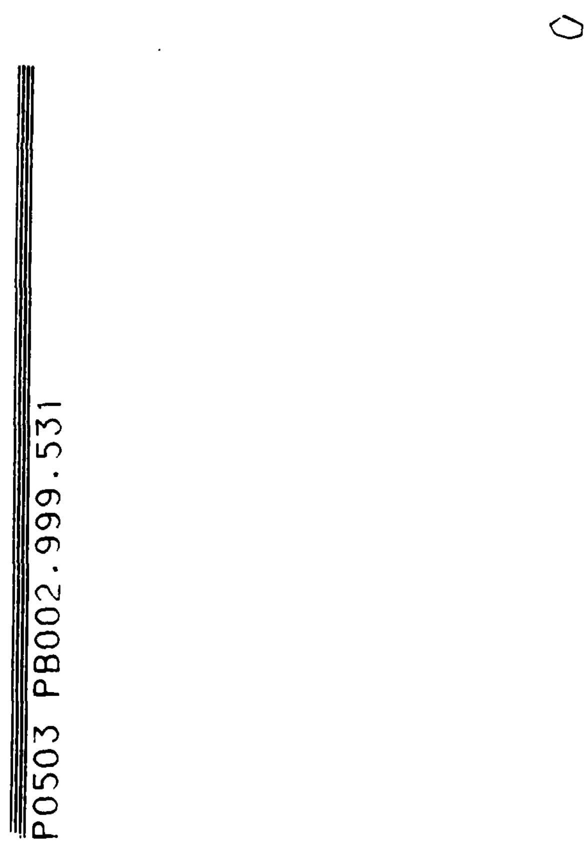


Figure 16 Isolated closed contours in Area 0

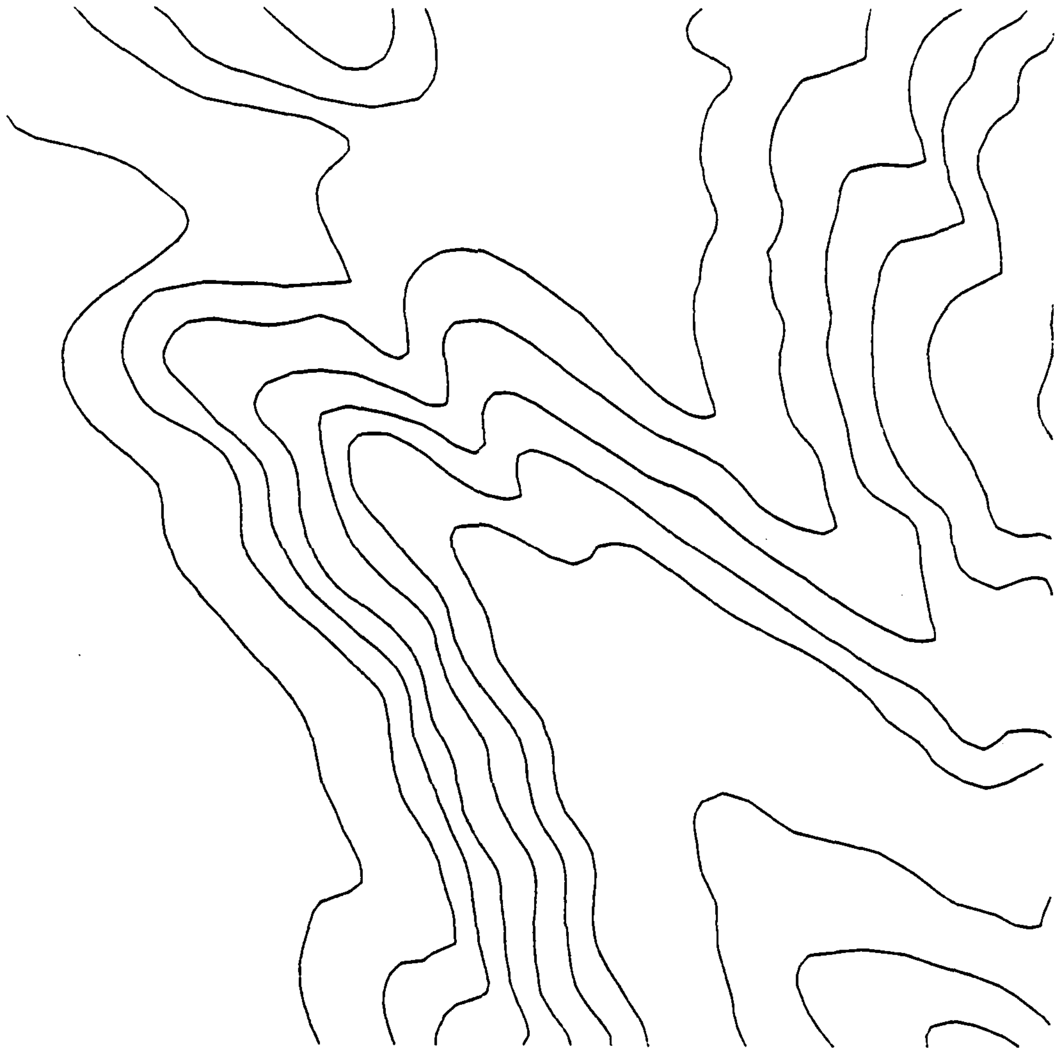


Figure 17   Contours which terminate on boundaries in Area 0

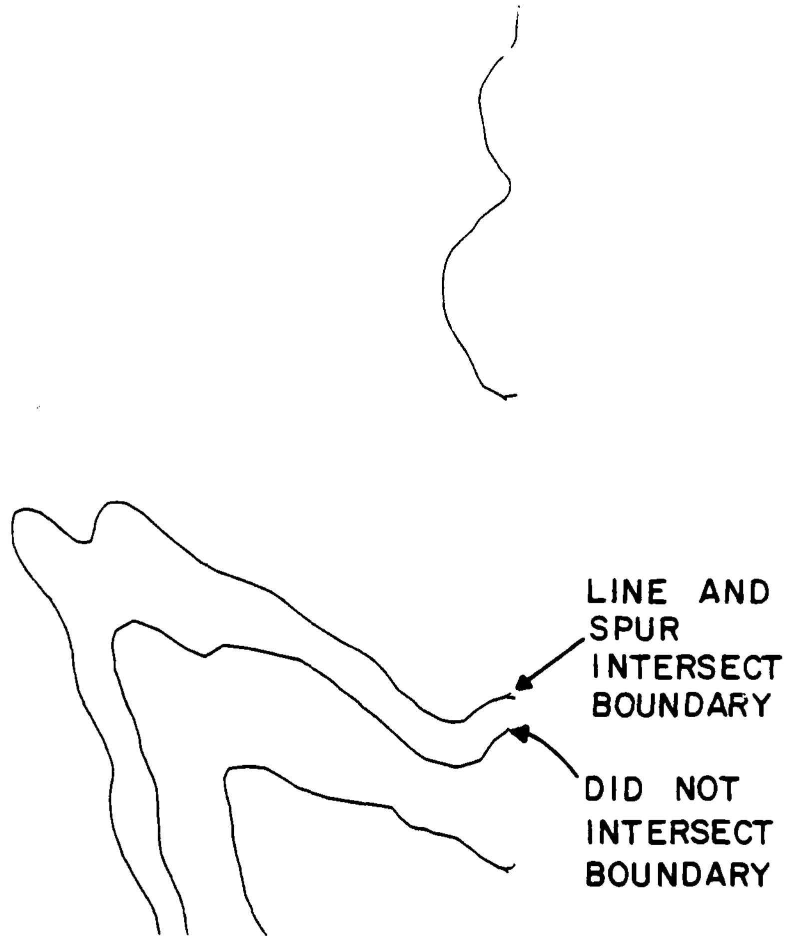


Figure 18 Contours requiring further editing in Area 0

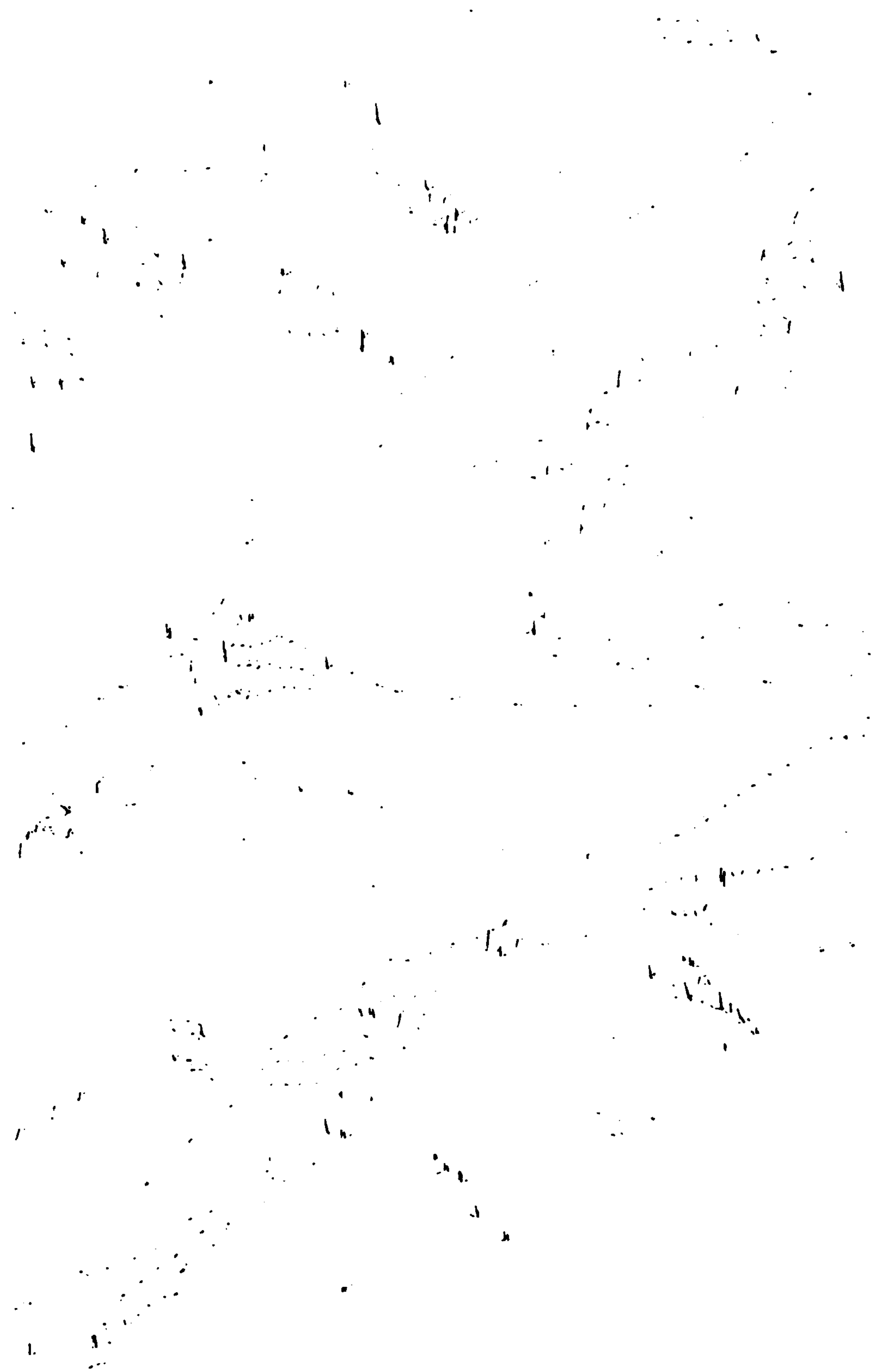


Figure 19   Spurs in Area 4

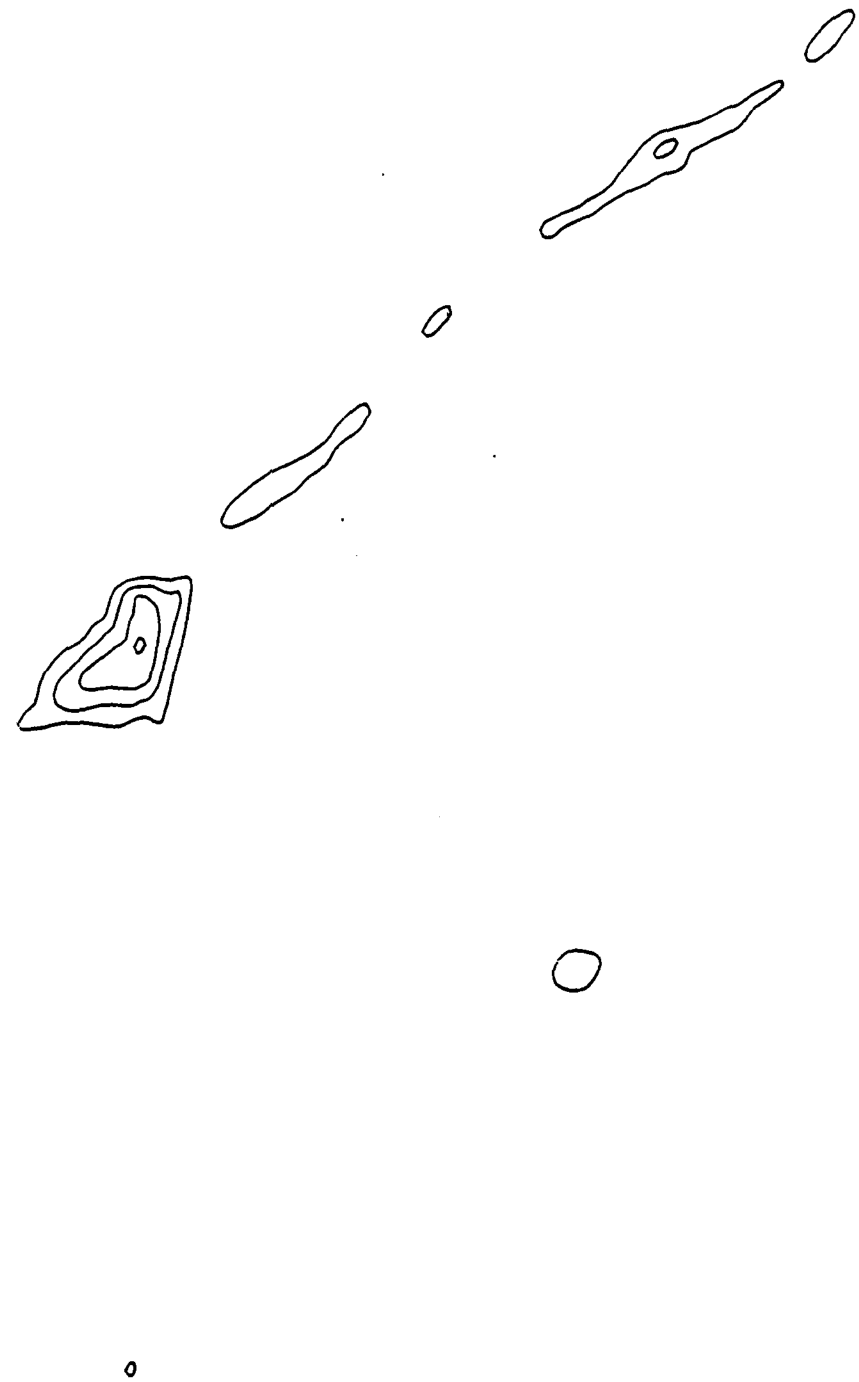


Figure 20   Closed contours in Area 4

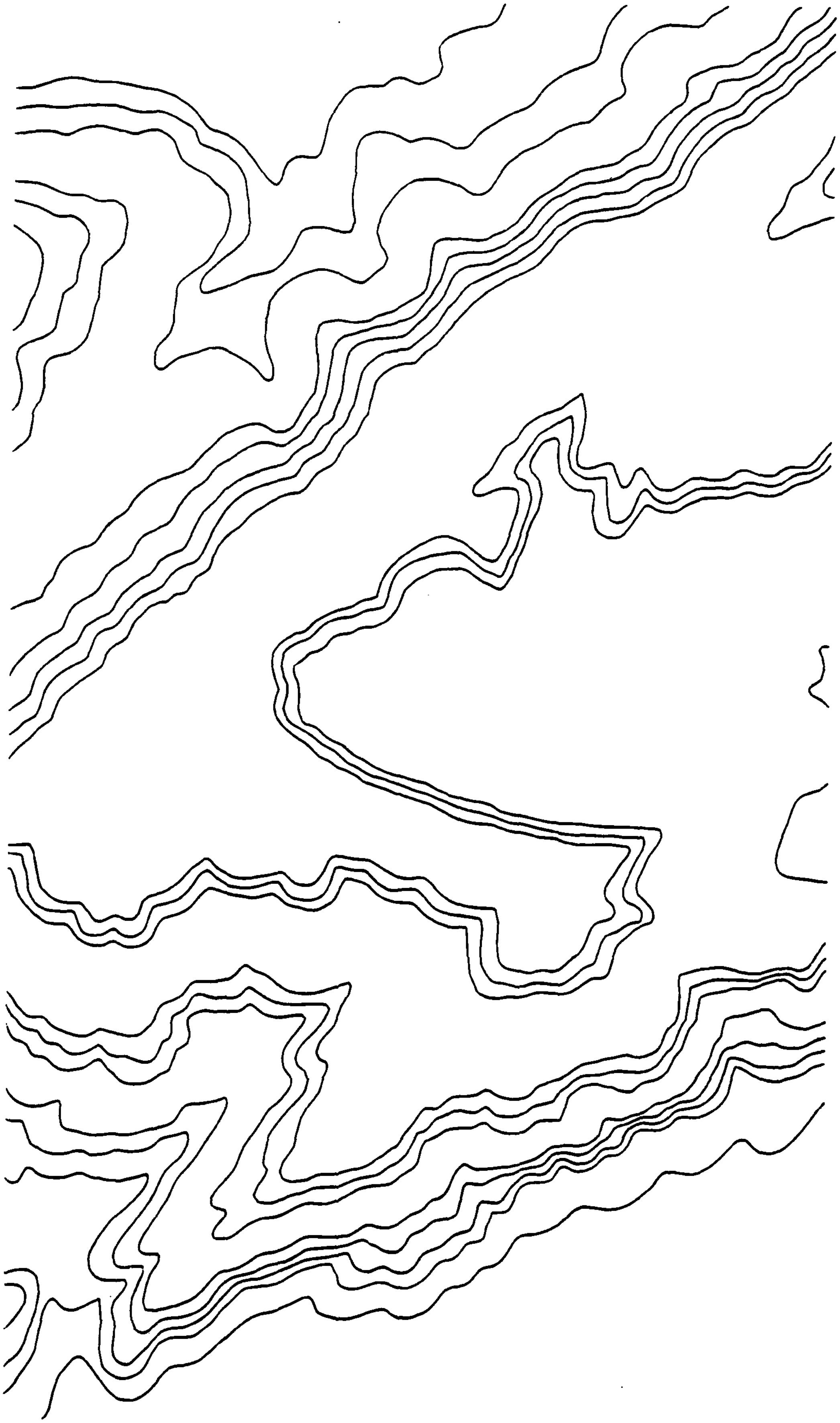


Figure 21 Contours which originate and terminate at boundaries in Area 4

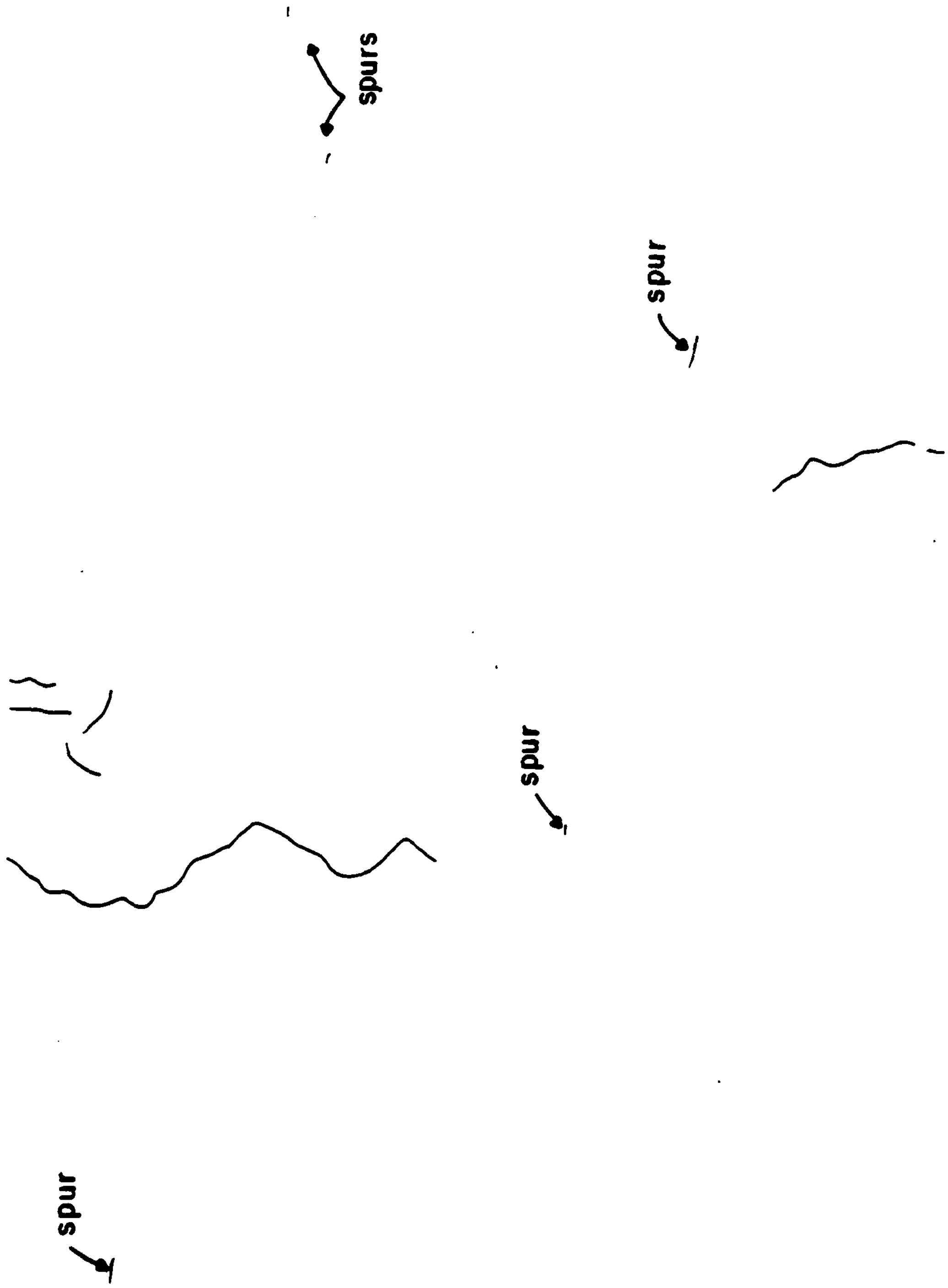


Figure 22 Open contours requiring further editing in Area 4

of which are longer contour lines which terminated because the contour line was broken either by design (see Figure 24) or by the scanner.

A number of contour lines required further editing as illustrated in Figure 23. The contour lines in this figure that intersect other contour lines (indicated on the figure), were open lines which were not labeled spurs, or were contour lines with spurs which intersected only one of two boundaries. Intersecting lines often occur in regions of high line density as illustrated by Area 2. The region on the right side of Figure 10 contained the broken and intersecting lines displayed in Figure 24. In this case, the scanner did not have sufficient resolution to separate the closely spaced lines. The original map may also have regions with broken lines where the line density was initially perceived as too high as illustrated in Figures 11 and 24 (Area 3).

Intersecting lines may occur in regions of high line density when the scanner does not have sufficient resolution to separate the lines, or may occur by design when a cut or fill is to be depicted. An example of a region with deliberate line intersection is given in Figures 9 and 26 (Area 1). In this case, further editing is required only if the application of the raster-to-vector processing includes the assignment of a height value to each contour line, in which case additional contour lines are required along the cut.

As illustrated by Figure 19, the mini raster-to-vector conversion program successfully detected over 700 spurs in Area 4, the largest test area, and identified the five index line spurs it could not automatically detect as open contours requiring manual editing (Figure 22). An improvement to the program which includes the automatic identification of index lines would enable a 100 percent detection rate by using different line length criteria for index lines and the interstitial contour lines.

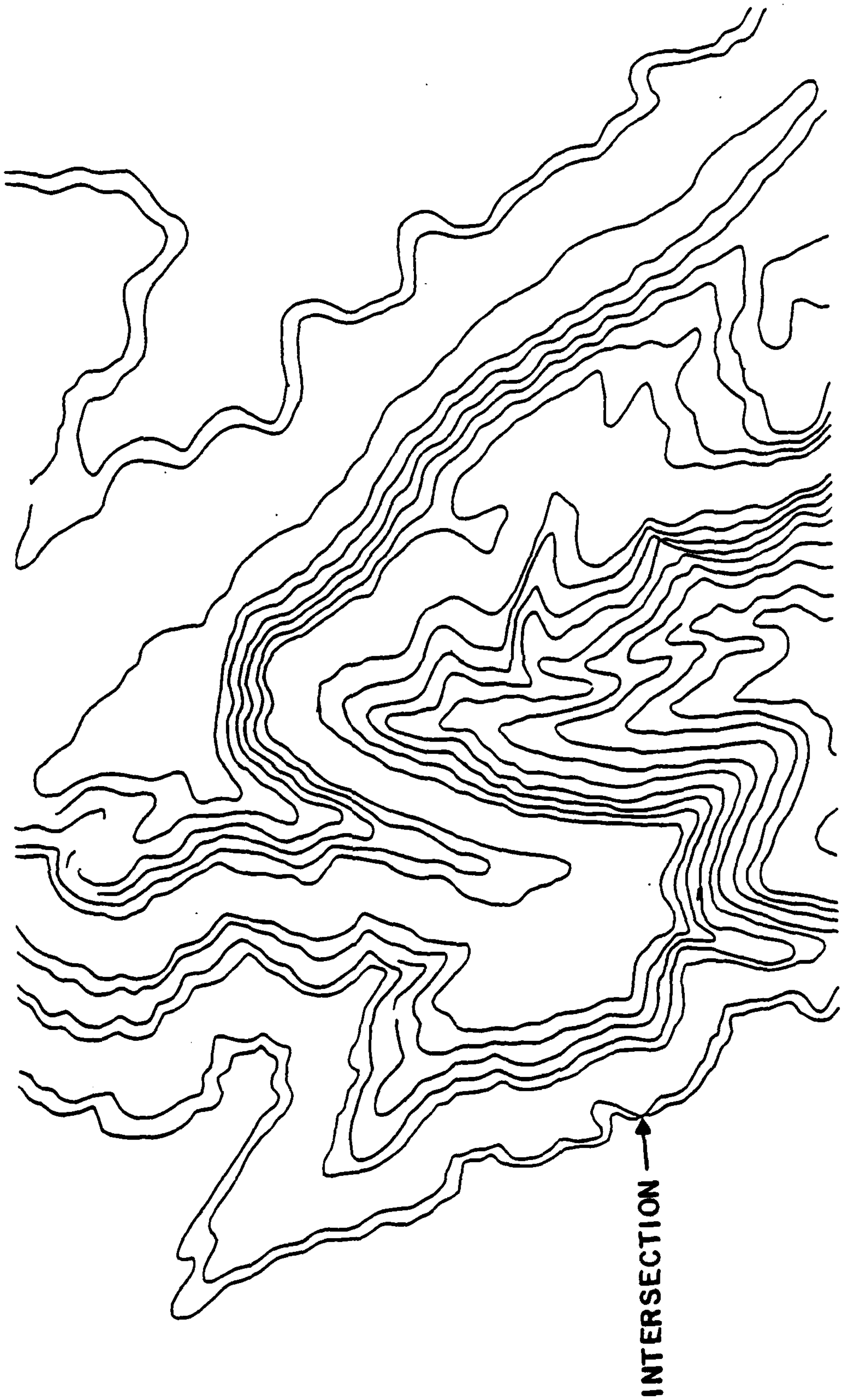


Figure 23 Intersecting contours requiring further editing in Area 4

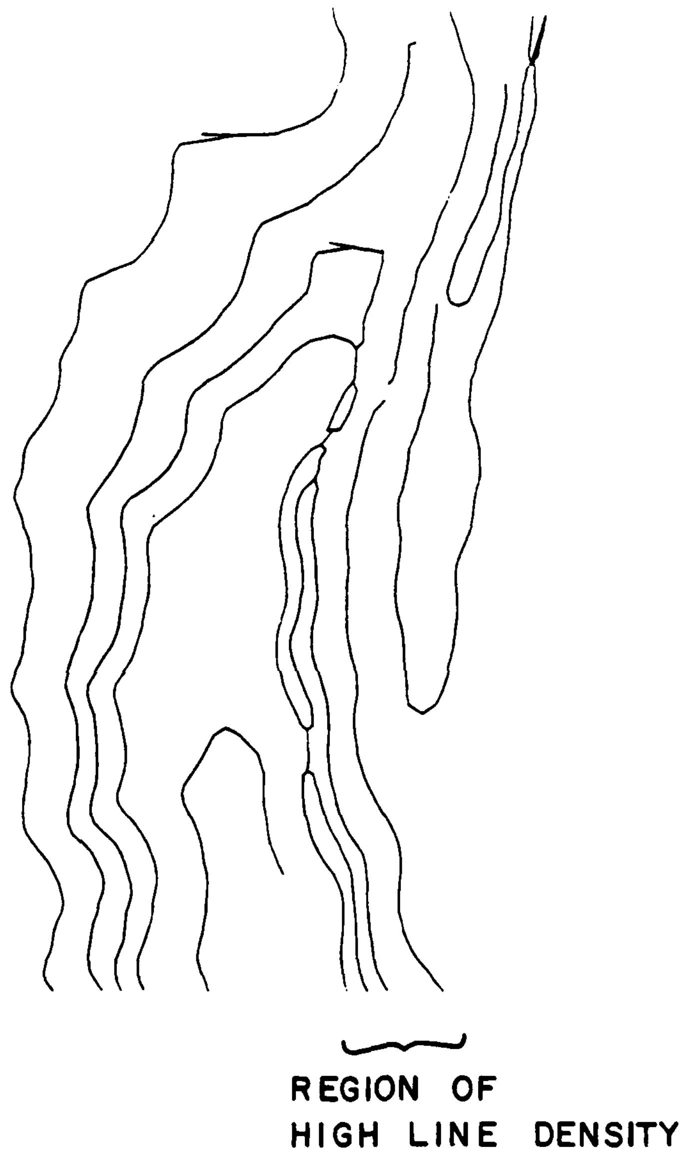


Figure 24 Contours requiring further editing in Area 2

Figure 25 Contour lines requiring further editing in Area 3

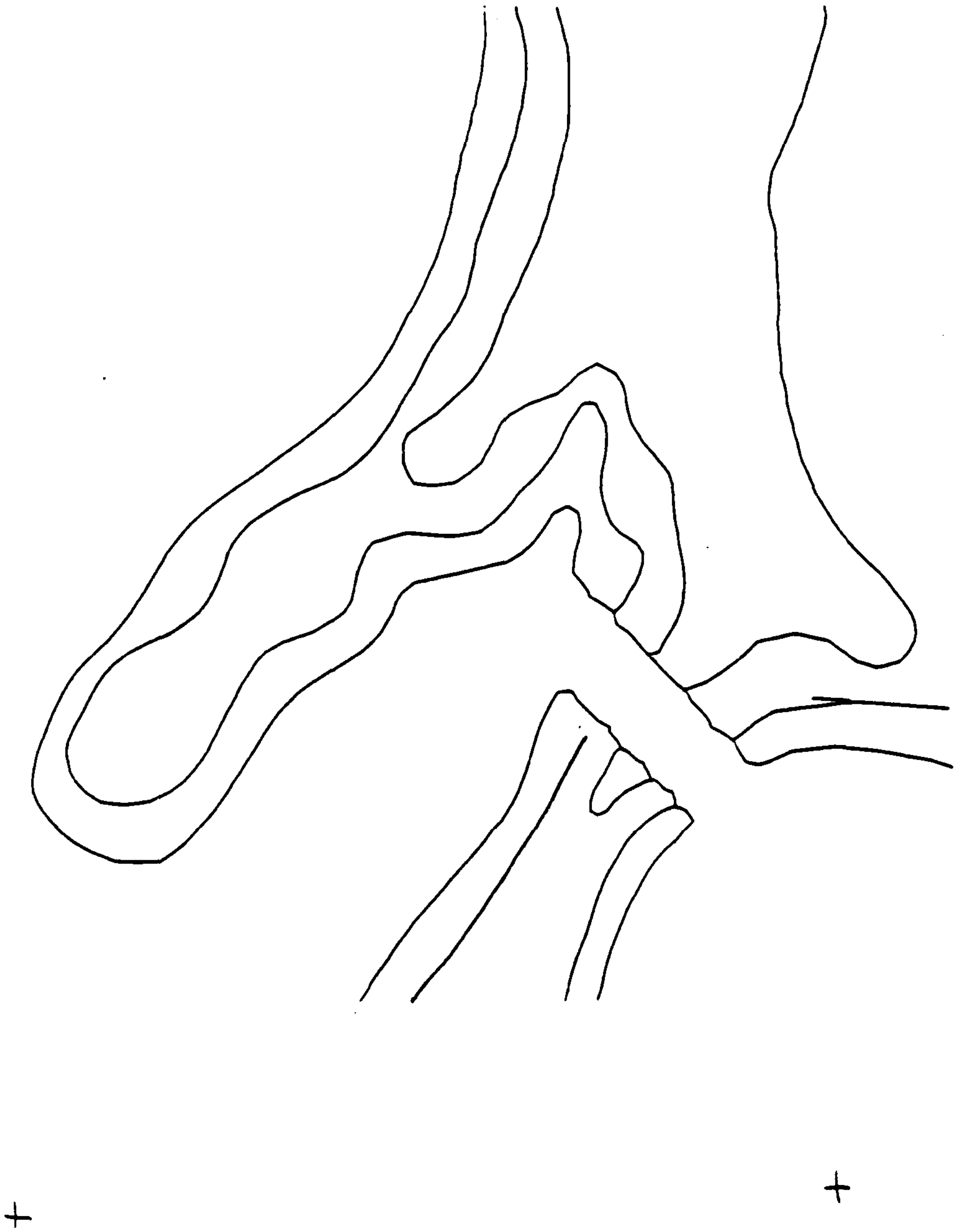


Figure 26   Contours requiring further editing in Area 1

4. RECOMMENDATIONS FOR AUTOMATED EDITING

The contour intersection (node) directories and contour line segment attributes were used to edit the data to prepare Figures 14 to 26. Many of the routine editing problems such as spur removal are readily solved using the contour line segment attributes. The directories can be used to determine how the line segments are interconnected to establish which contours follow all the rules for a topographic map and need no further editing (Figures 20 and 21 for example) and which contours need further editing (Figures 23, 24, 25 and 26 for examples). More automated editing is needed to significantly reduce the total time required to produce concatenated vector arrays for each contour and to label each contour with a height value.

A number of time-consuming editing operations currently performed manually at computer terminal work stations can be automated. The functions to be automated and the techniques used depend on the application of the digital cartographic operation. For example, consider the editing required to prepare two-dimensional grids (arrays) of height data from a vectorized contour map. First, the routine editing must be accomplished to close broken lines which result from scanner imperfections and errors in the manual map preparation for scanning. Second, the more subtle problems of separating overlayed contour lines on cuts or fills must be accomplished. Third, the heights of each contour line must be set. Fourth, the ridge and valley lines must be overlayed so that slopes can be estimated, and fifth, the height grid must be calculated using the slope values to interpolate between contour lines. These five steps can be accomplished automatically, starting from a multi-color scan of a topographic map that has not been specially prepared for raster-to-vector scanning and conversion.

The following steps are recommended to develop a rapid and completely automatic scheme for the preparation of gridded height data.

1) Finish the routine editing of a topographic map

    a) reinstate the thickness attribute and label index lines,

    b) connect the sparse set of broken index lines using proximity between line segments,

c) separate index lines from interstitual contour lines using thickness information as a guide,

d) connect broken interstitual lines using proximity to index lines,

2) Establish contour line nesting

a) introduce left, right neighbor attributes and directories for multiple neighbors,

b) establish high to low nesting order; each closed contour line is contained in one enclosing contour but may enclose several contours,

c) locate peaks and valleys; peaks may be separated from valleys by internal spurs,

d) establish the relative height of each contour from the nesting information,

e) fix the absolute height of each contour by the manual insertion of the absolute height of a map feature (index line, peak, valley, etc.),

f) from the nesting information complete the editing of deliberate contour intersections or breaks in contour lines,

3) Multi-color processing without map preparation

a) recognize contour lines (brown or blue on a topographic map),

b) recognize and remove overlaying information (black or red),

c) using nesting data, extend contour lines across overlaying information,

d) using nesting data, extend index lines and neighboring interstitual lines across index line labels (see Figure 11),

e) using pattern recognition techniques, recognize the height of the labeled index lines,

4) Preparation of gridded height data

a) establish slopes from vector data output prior to reordering (reordering is necessary only if the data are stored in the AGDS format),

b) construct height arrays from vector data using the nesting information (ridge and valley lines should not be necessary).

The ability of DMA10 to carry additional attributes as suggested above will require more core storage than is available on the DMAHTC PDP 11/60 minicomputer. Two options are available to overcome this limitation; (1) a different computer with more available core storage could be utilized or (2) the map could be partitioned and then analyzed over more than one pass. The second option would have the added benefit of significantly increasing the speed of the reordering routine, FETCH5. DMA10 stores contour segments, nodes and attributes sequentially onto disk as it encounters them in the data. While each disk record is tagged with a contour identifier, the logic in FETCH5 must search the entire disk for each successive record that together form a full contour line. By partitioning the data, the region of disk that must be searched is reduced thereby minimizing the disk search time. However, the time required for contouring will be largely unaffected since DMA10 will continue to process the map data as it is encountered, only the time required to read through the data to the partition point will be added.

REFERENCES

Crane, R.K., 1979: "Automatic Cell Detection and Tracking", IEEE Trans. Geoscience Elect., GE-17, 250-262.

Gustafson, G.B., 1981: "Documentation of a Mini Raster-to-Vector Conversion Program", ERT Doc. No. P-B002-T, Environmental Research & Technology, Inc., Concord, Mass.


8